encryption or authentication key.

43. (New)The method as claimed in Claim 37 wherein individual processors add result data to the control data.

44. (New)The method as claimed in Claim 37 wherein the processors perform IPSEC protocal processing.

45. (New)The method as claimed in Claim 37 wherein respective processors perform IP header manipulation and encryption.

46. (New)The method as claimed in Claim 45 wherein a processor performs authentication processing.

REMARKS

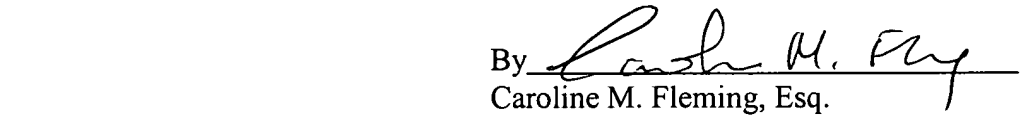
Claims 1-27 are pending. Claims 28-46 have been added. The specification has been amended to add a description of the buffer controller. Support for the amendment is provided in the application as originally filed. (Page 6, lines 1-6 and Claim 1, lines 11-13.) The specification has also been amended to correct typographical errors. A proposed drawing correction is submitted concurrently herewith for Figs. 1, 3, 4 and 10. No new matter has been added.

CONCLUSION

In view of the above amendments and remarks, it is believed that all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned at (978) 341-0036.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By 

Caroline M. Fleming, Esq.
Registration No. 45,566
Telephone: (978) 341-0036
Facsimile: (978) 341-0136

Concord, MA 01742-9133
Dated: 7/23/02

MARKED UP VERSION OF AMENDMENTS

Specification Amendments Under 37 C.F.R. § 1.121(b)(1)(iii)

Replace the paragraph at page 9, lines 5 through 22 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph.

Referring to Fig. 3, a simplified architectural diagram of an embodiment of the present invention is shown. Here a data buffer 30 is shown disposed central to a packet processor. A buffer controller 31 determines based on the header within a packet, a next processor of a plurality of processors to process data packets. The buffer controller 31 provides the data packet to a port for provision to the next processor. A master processor 32 acts to format each packet in order to insert a header therein indicative of processes required for processing that packet. The master processor 32 is programmable and understands the processing of packets at a high level. Once the packet is reformatted, it is returned to the data buffer 30 from which it is routed to a processing element 34 for performing the first listed function. For example, in the example of Fig. 2, the first function is determining a format of the packet. The packet format is determined and for each determined format a number of possible functions may be added or removed from the list within the header. For example, an encrypted packet may have the function cipher added to it along with some form of key identifier. The key identifier and the packet is then provided to a cipher processor from the buffer 30. In the cipher processor the packet is decrypted and the decrypted packet is returned to the buffer 30. The buffer 30 continues to provide the packet to processors as long as further functions remain within the header. When the header is empty, the packet is transferred to an output port for storage, for example in a received data buffer 36. Alternatively, a last function indicates the provision of the data to a data output port.

Replace the paragraph at page 14, lines 16 through 26 with the below paragraph marked up by way of bracketing and underlining to show the changes relative to the previous version of the paragraph.

In accordance with another embodiment of the invention as shown in Fig. 10, the server processor 106 stores within the header switching information for use in switching the super packet within an array of processors 106. A packet is directed [fro] from the server processor 106 to a first processor 100 for processing. The header and the packet data are separated so as to not affect processing of the data. When the data is processed, header data is provided to an output addressing switch 104 and the super packet data is automatically routed in a pseudo pipelined fashion to a subsequent processing element. Such an embodiment reduces flexibility, expandability, functionality and so forth while adding [tot he] to the overall hardware complexity. That said, the performance of such an embodiment is likely superior to the more flexible architecture described above and in many applications the lack of flexibility and so forth is not considered a great disadvantage.